CMPT385 Final Exam

Three Hours
Closed Book. No calculators or computers.
December 13, 2006, 2.00pm

Instructions:

Read each question before you answer it. Marks cannot be awarded if you answer some other question.
Indicate your answers clearly.
Please do not use a hard pencil.
Marks will be awarded for the style and clarity of your answer.

1. **Graphics Pipeline**

- matrices
- [illegible]

1.1. **(6 marks)** Identify three OpenGL state variables, and the commands used to change them.

1.2. **(10 marks)** Define in a sentence each of the following operations in the OpenGL Pipeline: depth calculation, lighting calculation, camera transformation, modelview calculation, scanline rendering. Then give the order that these operations are done.

1.3. **(7 marks)** Explain implementing realistic shadows is a problem in the OpenGL pipeline, and describe one simple way to implement approximate shadows in OpenGL.

2. **Geometric Transformations**

For the following, assume a two-dimensional graphics system. and that points are represented as column vectors. Thus, the modelview matrix is initially a 3x3 identity matrix, and calls to *glTranslate()* etc., result in a *right* multiplication of the modelview matrix by a transformation matrix.

(Assume access to a library to compute any of the usual math functions.)

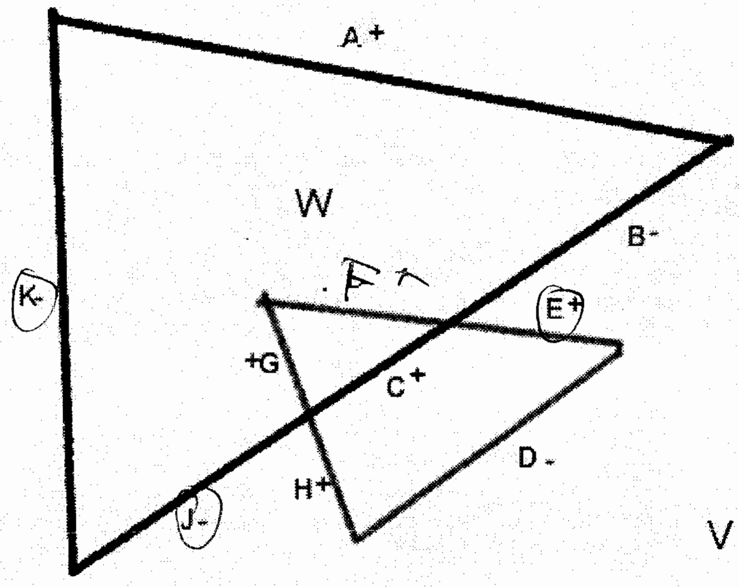
2.1. **(8 marks)** The bottom row of the modelview matrix is initially (0, 0, 1). Do the values in this row ever change as a consequence of *translate, rotate* or *scale* operations? Show why.

2.2. **(4 marks)** An application of a *translate* operation to *any* modelview matrix representing a sequence of *translate, rotate* and *scale* operations affects how many entries of the modelview matrix? Show your work.

2.3. **(4 marks)** Show that any modelview matrix consisting of an arbitrary sequence of rotations and translations can be reduced to a single rotation and translation.

3. **Lighting**
   3.1. **(9 marks)** OpenGL uses Gouraud lighting. Summarize this lighting model and explain why it is used in OpenGL.
   3.2. **(6 marks)** Briefly describe some other lighting model (radiosity, photon mapping, Phong) and discuss the feasibility of implementing these in real time using an OpenGL-like implementation.

4. **Depth Calculations**
   4.1. **(4 marks)** Explain the z-buffer algorithm.
   4.2. **(12 marks)** Consider the following scene similar to the examples used in class.



The drawing represents a bird's eye view of several rectangles arranged in two triangular shapes which intersect. The rectangles are subdivided at the intersection points, and each rectangle is labeled with a letter. Adjacent to the letter is a +/- sign indicating that the equation of that rectangle's plane is +/- in the vicinity of the letter. Show the BSP tree that would result if the following rectangles are added in the following order:

F B C G A H D

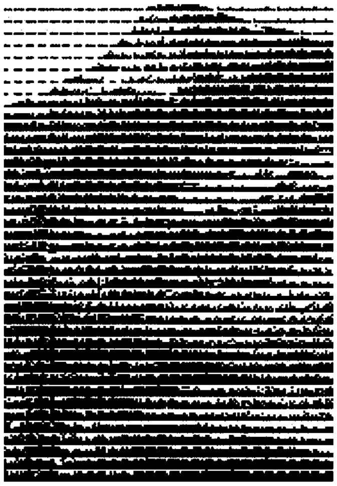
   4.3. **(8 marks)** Show the order the above rectangles would be rendered from viewpoints *V* and *W* respectively.
   4.4. **(Bonus)** Why didn't I include ~~J and~~ K in the above list?

DO EITHER QUESTION 5 OR QUESTION 6

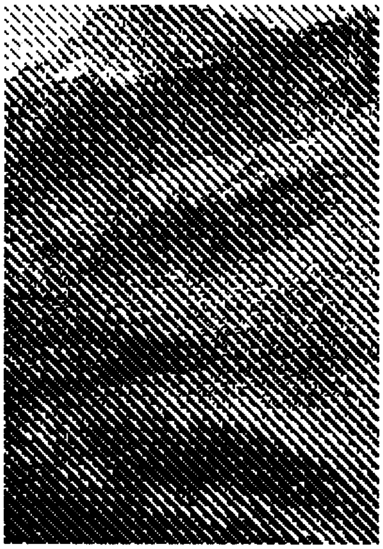
5. **Halftoning** (10 marks)

The following matrix could be used as a dither matrix for the halftoning assignment you did in this class. It is called a "line halftone" and creates the output shown beside the data:

```
{{36, 35, 34, 33, 32, 31},
{30, 29, 28, 27, 26, 25},
{24, 23, 22, 21, 20, 19},
{18, 17, 16, 15, 14, 13},
{12, 11, 10,  9,  8,  7},
{ 6,  5,  4,  3,  2,  1}};
```



The dither matrix above is very similar to the line halftone matrix used in your assignment. It is called a line matrix because there is an effect of lines crossing the image. Show how to create a line matrix that creates a diagonal line effect as follows:



DO EITHER QUESTION 5 OR QUESTION 6

6. **Shaders**
   - 6.1. **(3 marks)** Explain the difference between surface shaders, bump maps, and displacement shaders.
   - 6.2. **(7 marks)** Describe how you would use Perlin noise to write a surface shader to create a mottled surface, as on an orange. (If you can write the shader, do so. If not, sketch out an idea.

7. **(12 marks) Short answer**
   - 7.1. Describe either particle systems or L-systems.
   - 7.2. What is picking, and how does OpenGL help with this?
   - 7.3. What is double buffering?
   - 7.4. What is the purpose of glPushMatrix() and glPopMatrix()?
   - 7.5. What is the difference between the modelview matrix and the projection matrix?
   - 7.6. Distinguish between geometric and parametric continuity.

End of exam